

合肥大學  
HEFEI UNIVERSITY



# Programming with Python

## 1. Organisation

Thomas Weise (汤卫思)  
[tweise@hfuu.edu.cn](mailto:tweise@hfuu.edu.cn)

School of Artificial Intelligence and Big Data  
Hefei University  
Hefei, Anhui, China

人工智能与大数据学院  
合肥大学  
中国安徽省合肥市

# Programming with Python



Dies ist ein Kurs über das Programmieren mit der Programmiersprache Python an der Universität Hefei (合肥大学).

Die Webseite mit dem Lehrmaterial dieses Kurses ist <https://thomasweise.github.io/programmingWithPython> (siehe auch den QR-Kode unten rechts). Dort können Sie das Kursbuch (in Englisch) und diese Slides finden. Das Repository mit den Beispielprogrammen in Python finden Sie unter <https://github.com/thomasWeise/programmingWithPythonCode>.



# Outline

1. Willkommen
2. Lehrer
3. Zeitplan
4. Aspekte
5. Lehrmaterial
6. Ökosystem
7. Voraussetzungen
8. Warnung
9. Zusammenfassung





Willkommen



# Willkommen

- Willkommen zum Kurs **Programmieren mit Python**.



# Willkommen



- Willkommen zum Kurs **Programmieren mit Python**.
- Wir befinden uns im Frühling/Sommer-Semester 2026 an der Universität Hefei (合肥大学) in der schönen Stadt Hefei (合肥市) in der Provinz Anhui (安徽省) in China.

# Willkommen



- Willkommen zum Kurs **Programmieren mit Python**.
- Wir befinden uns im Frühling/Sommer-Semester 2026 an der Universität Hefei (合肥大学) in der schönen Stadt Hefei (合肥市) in der Provinz Anhui (安徽省) in China.
- Dieser Kurs lehrt das Programmieren mit der Programmiersprache Python.

# Willkommen



- Willkommen zum Kurs **Programmieren mit Python**.
- Wir befinden uns im Frühling/Sommer-Semester 2026 an der Universität Hefei (合肥大学) in der schönen Stadt Hefei (合肥市) in der Provinz Anhui (安徽省) in China.
- Dieser Kurs lehrt das Programmieren mit der Programmiersprache Python.
- Der Kurs wird in deutscher Sprache unterrichtet.



Lehrer



# Lehrer

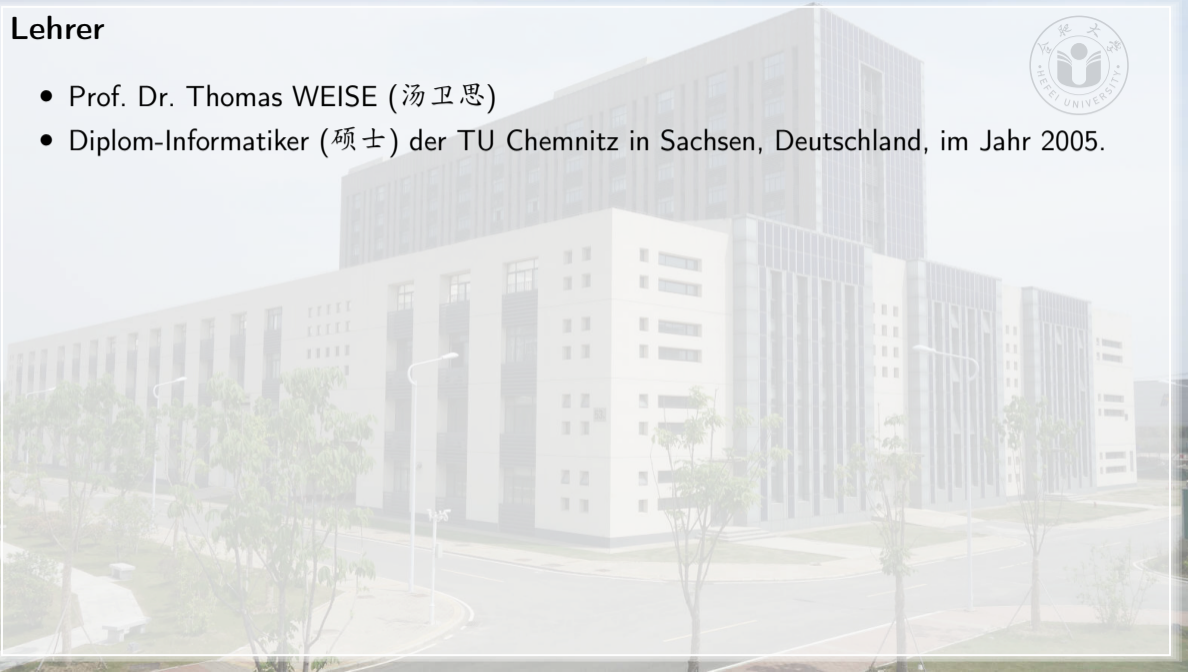
- Prof. Dr. Thomas WEISE (汤卫思)



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



© Ewald Juddt CC BY 4.0



© Kolossos CC BY-SA 3.0

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



© Ewald Judt CC BY 4.0



© Ewald Judt CC BY 4.0



© Kolossos CC BY-SA 3.0

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



© Ewald Judt CC BY 4.0



© Ewald Judt CC BY 4.0



© Kolossos CC BY-SA 3.0

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



© Ewald Judt CC BY 4.0



© eagle77 Pixabay Content License



© Kolossos CC BY-SA 3.0

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



© Ewald Judt CC BY 4.0



© eagle77 Pixabay Content License



© Kolossos CC BY-SA 3.0

© eagle77 Pixabay Content License

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



© Ewald Judt CC BY 4.0



© eagle77 Pixabay Content License

© Firstclasspixel Pixabay Content License

© Kolossos CC BY-SA 3.0

© eagle77 Pixabay Content License

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



© Ewald Judt CC BY 4.0



src, copyright Jacob Müller, not Creative Commons, mit persönlicher Genehmigung, see also TU Chemnitz Bilddatenbank

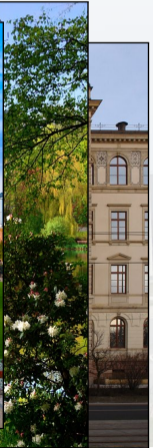
© eagle77 Pixabay Content License

© Firstclasspixel Pixabay Content License



© Kolossos CC BY-SA 3.0

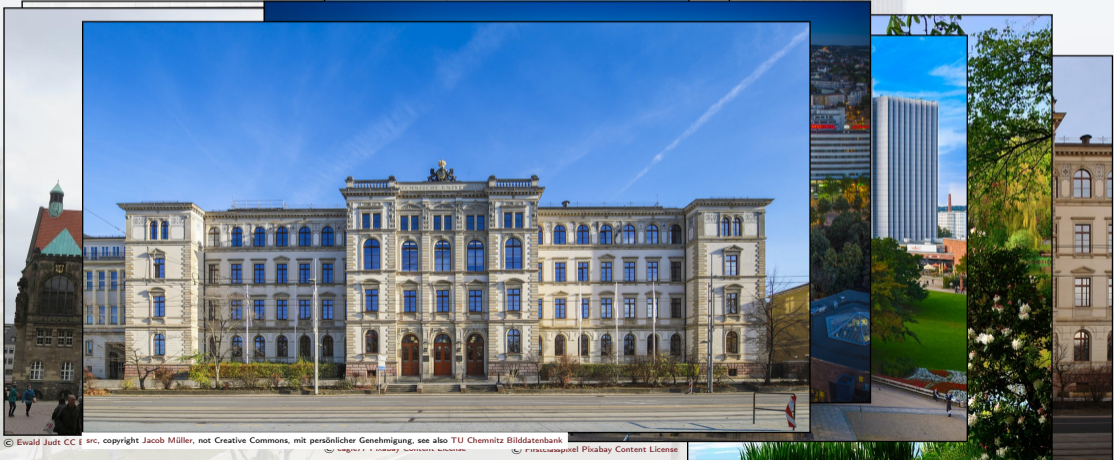
© eagle77 Pixabay Content License



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.

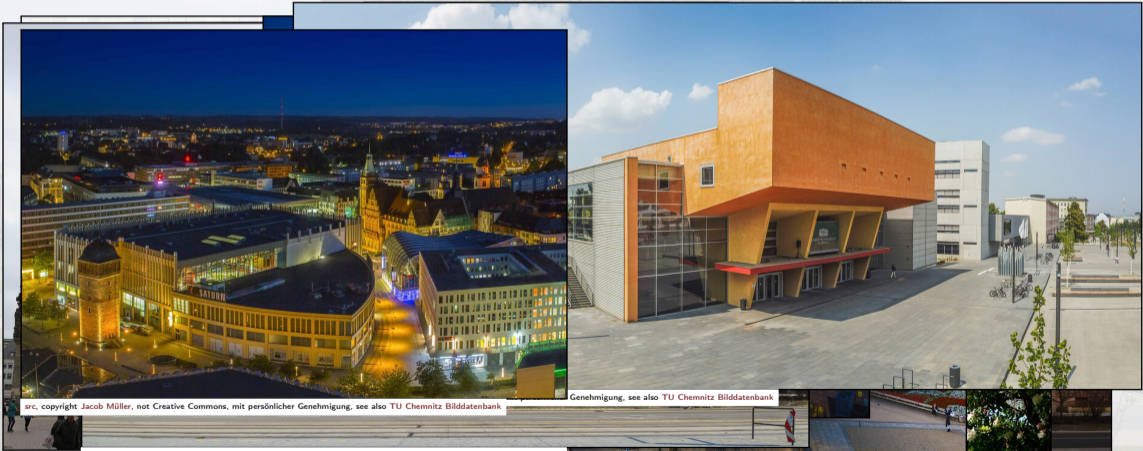


src, copyright Jacob Müller, not Creative Commons, mit persönlicher Genehmigung, see also TU Chemnitz Bilddatenbank

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



src, copyright Jacob Müller, not Creative Commons, mit persönlicher Genehmigung, see also TU Chemnitz Bilddatenbank

Genehmigung, see also TU Chemnitz Bilddatenbank

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.



src, copyright Jacob Müller, not Creative Commons, mit persönlicher Genehmigung, see also TU Chemnitz Bilddatenbank



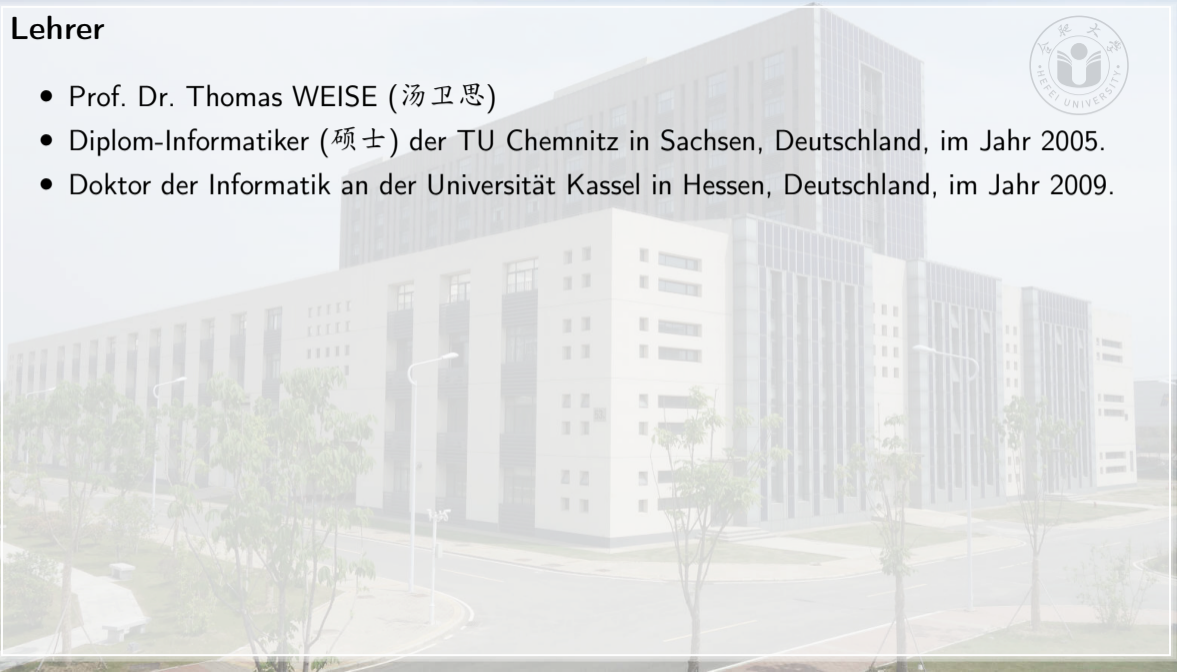
src, copyright Jacob Müller, not Creative Commons, mit persönlicher Genehmigung, see also TU Chemnitz Bilddatenbank

Genehmigung, see also TU Chemnitz Bilddatenbank

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



© till\_westhof Pixabay Content License



© webandi Pixabay Content License

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



© till\_westhof Pixabay Content License



© webandi Pixabay Content License



© Barni1 Pixabay Content License

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



© till\_westhof Pixabay Content License



© webar

© Barni1 Pixabay Content License

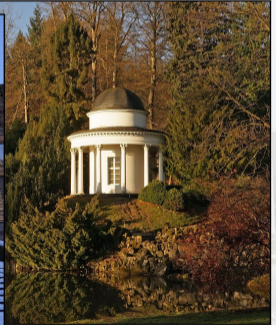


ense

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



U N I K A S S E L  
V E R S I T Ä T

Copyright Universität Kassel, mit Genehmigung der Präseskanzlei der Universität Kassel  
© till\_westhof Pixabay Content License

© webar

© Barni1 Pixabay Content License

ense

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



U N I K A S S E L  
V E R S I T Ä T

Copyright Universität Kassel, mit Genehmigung der Pressestelle der Universität Kassel  
© till\_westhof Pixabay Content License

© webar

© Hiltz/Stockbyte/Alamy

© Barni1 Pixabay Content License

Copyright Universität Kassel, mit Genehmigung der Pressestelle der Universität Kassel

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



UNI KASS  
VERSITÄ

Copyright Universität Kassel, mit Genehmigung der Pressestelle der Universität  
© till\_westhof Pixabay Content License

Copyright Universität Kassel, mit Genehmigung der Pressestelle der Universität Kassel

© webar

© Barni1 Pixabay Content License

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



U N I K A S S  
V E R S I T Ä

Copyright Universität Kassel, mit Genehmigung der Pressestelle der Universität Kassel  
© till\_westhof Pixabay Content License

Copyright Universität Kassel, mit Genehmigung der



© webar  
© Barn1 P

Copyright Universität Kassel, mit Genehmigung der Pressestelle der Universität Kassel

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.



U N I K A  
V E R S

Copyright Universität Kassel, mit Genehmigung der  
© till\_westhof Pixabay Content License



Copyright Universität Kassel, mit Genehmigung der Pressestelle der Universität Kassel



g der Pressestelle der Universität Kassel

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.



# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.
- 2011-2016: Associate Professor (副教授) an der USTC.

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.
- 2011-2016: Associate Professor (副教授) an der USTC.
- Seit 2016: Full Professor (教授) und Direktor des Institute of Applied Optimization (应用优化研究所) der Universität Hefei (合肥大学) in Hefei, Anhui, China.

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.
- 2011-2016: Associate Professor (副教授) an der USTC.
- Seit 2016: Full Professor (教授) und Direktor des Institute of Applied Optimization (应用优化研究所) der Universität Hefei (合肥大学) in Hefei, Anhui, China.
- Forschungsfeld: Optimierung, Operations Research.

# Lehrer



- Prof. Dr. Thomas WEISE (汤卫思)
- Diplom-Informatiker (硕士) der TU Chemnitz in Sachsen, Deutschland, im Jahr 2005.
- Doktor der Informatik an der Universität Kassel in Hessen, Deutschland, im Jahr 2009.
- 2009-2011: PostDoc (博士后) an der University of Science and Technology of China (USTC, 中国科学技术大学) in Hefei, Anhui, China.
- 2011-2016: Associate Professor (副教授) an der USTC.
- Seit 2016: Full Professor (教授) und Direktor des Institute of Applied Optimization (应用优化研究所) der Universität Hefei (合肥大学) in Hefei, Anhui, China.
- Forschungsfeld: Optimierung, Operations Research.
- Kontakt: [tweise@hfuu.edu.cn](mailto:tweise@hfuu.edu.cn).



# Zeitplan



# Zeitplan: Python 2025/2026 2



Vorlesung						Vorlesung / Übung (36-503)					
周次	月-日	星期	节	Zeit	Raum	周次	月-日	星期	节	Zeit	Raum
5	03-30	Mo	3-4	10:20-12:00	36-204	5	04-01	Mi	3-4	10:20-12:00	36-204
						6	04-08	Mi	3-4	10:20-12:00	36-204
7	04-13	Mo	3-4	10:20-12:00	36-204	7	04-15	Mi	3-4	10:20-12:00	36-204
						7	04-17	Fr	3-4	10:20-12:00	36-503
8	04-20	Mo	3-4	10:20-12:00	36-204	8	04-22	Mi	3-4	10:20-12:00	36-204
						8	04-24	Fr	3-4	10:20-12:00	36-503
9	04-27	Mo	3-4	10:20-12:00	36-204	9	04-29	Mi	3-4	10:20-12:00	36-204
10	05-04	Mo	3-4	10:20-12:00	36-204	10	05-06	Mi	3-4	10:20-12:00	36-503
12	05-18	Mo	3-4	10:20-12:00	36-206	12	05-20	Mi	3-4	10:20-12:00	36-503
13	05-25	Mo	3-4	10:20-12:00	36-206	13	05-27	Mi	3-4	10:20-12:00	36-503
14	06-01	Mo	3-4	10:20-12:00	36-206	14	06-03	Mi	3-4	10:20-12:00	36-503
15	06-08	Mo	3-4	10:20-12:00	36-206	15	06-10	Mi	3-4	10:20-12:00	36-503
16	06-15	Mo	3-4	10:20-12:00	36-206	16	06-17	Mi	3-4	10:20-12:00	36-503



# Aspekte



# Aspekte



# Aspekte



- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite



- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite:
  1. Alles, was gelehrt wird, wird aktiv praktisch und selbstständig von den Studierenden ausprobiert.



- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite:
  1. Alles, was gelehrt wird, wird aktiv praktisch und selbstständig von den Studierenden ausprobiert. **Auf Ihrem eigenen Laptop-Computer.**



- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite:
  1. Alles, was gelehrt wird, wird aktiv praktisch und selbstständig von den Studierenden ausprobiert. **Auf Ihrem eigenen Laptop-Computer.**
  2. Umfangreiches unterstützendes Lehrmaterial wird zur Verfügung gestellt.



- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite:
  1. Alles, was gelehrt wird, wird aktiv praktisch und selbstständig von den Studierenden ausprobiert. **Auf Ihrem eigenen Laptop-Computer.**
  2. Umfangreiches unterstützendes Lehrmaterial wird zur Verfügung gestellt. Alles kann nachgelesen und nachvollzogen werden.



- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite:
  1. Alles, was gelehrt wird, wird aktiv praktisch und selbstständig von den Studierenden ausprobiert. **Auf Ihrem eigenen Laptop-Computer.**
  2. Umfangreiches unterstützendes Lehrmaterial wird zur Verfügung gestellt. Alles kann nachgelesen und nachvollzogen werden.
  3. Das ist keine Arbeit, kein Auswendiglernen.



- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite:
  1. Alles, was gelehrt wird, wird aktiv praktisch und selbstständig von den Studierenden ausprobiert. **Auf Ihrem eigenen Laptop-Computer.**
  2. Umfangreiches unterstützendes Lehrmaterial wird zur Verfügung gestellt. Alles kann nachgelesen und nachvollzogen werden.
  3. Das ist keine Arbeit, kein Auswendiglernen. Es soll nicht langweilig sein.



- Die Durchführung des Kurses hat drei wesentliche Aspekte von meiner Seite:
  1. Alles, was gelehrt wird, wird aktiv praktisch und selbstständig von den Studierenden ausprobiert. **Auf Ihrem eigenen Laptop-Computer.**
  2. Umfangreiches unterstützendes Lehrmaterial wird zur Verfügung gestellt. Alles kann nachgelesen und nachvollzogen werden.
  3. Das ist keine Arbeit, kein Auswendiglernen. Es soll nicht langweilig sein. Das Ziel ist, dass Sie Freude beim Lernen und bei der Arbeit mit unserem Themengebiet fühlen.



# Lehrmaterial



# Lehrmaterial





Programming with Python x +

thomasweise.github.io/programmingWithPython/

## Programming with Python

- 1. Introduction
- 2. Resources
  - 2.1. The Book
  - 2.2. Die Slides in Deutsch — The Slides in German Language
  - 2.3. The Slides in English
  - 2.4. The Example Python Programs
  - 2.5. LaTeX Sources, Graphics, Bibliography, etc.
- 3. License
- 4. Contact

### 1. Introduction

*"Programming with Python"* is both a book and a course held at [Hefei University \(合肥大学\)](#). We here provide all the resources of this course.

The goal of the course and book is to teach undergraduate and graduate students how to program with the Python programming language. The resources may also be suitable to high school students or whoever else may be interested in the topic. Our aim is to strike a good balance between theory and practice, leaning more to the practice side. In particular, we try to teach programming together with some software engineering concepts. It is the firm opinion of the author that these two cannot be separated. Teaching programming alone without introducing tools such as static code analysis, unit tests, and enforcing principles such as code style and proper commenting will create bad programmers. So we discuss these aspects while working our way through the principles of programming.

### 2. Resources

The website with up-to-date information about the course is <https://thomasweise.github.io/>



Programming with Python x +

thomasweise.github.io/programmingWithPython/

alone without introducing tools such as static code analysis, unit tests, and enforcing principles such as code style and proper commenting will create bad programmers. So we discuss these aspects while working our way through the principles of programming.

## 2. Resources

---

The website with up-to-date information about the course is <https://thomasweise.github.io/programmingWithPython/>.

### 2.1. The Book

---



The book itself can be downloaded as pdf at <https://thomasweise.github.io/programmingWithPython/programmingWithPython.pdf>.

### 2.2. Die Slides in Deutsch — The Slides in German Language

---

Die Slides zum Kurs in deutscher Sprache können unter <https://thomasweise.github.io/programmingWithPythonSlidesDE> gefunden werden und sind hier aufgelistet:

1. Organisation
2. Einleitung
3. Python Installieren
4. PyCharm Installieren
5. Programme Erstellen und Ausführen
6. Beispiele Herunterladen
7. Der Datentyp `int`



# Buch



- Ich habe für diesen Kurs ein Buch geschrieben: *Programming with Python*<sup>80</sup>.



# Buch



- Ich habe für diesen Kurs ein Buch geschrieben: *Programming with Python*<sup>80</sup>.
- Das Buch ist in Englisch geschrieben.



# Buch



- Ich habe für diesen Kurs ein Buch geschrieben: *Programming with Python*<sup>80</sup>.
- Das Buch ist in Englisch geschrieben.
- Dieses Buch steht kostenlos im Internet.

# Buch



- Ich habe für diesen Kurs ein Buch geschrieben: *Programming with Python*<sup>80</sup>.
- Das Buch ist in Englisch geschrieben.
- Dieses Buch steht kostenlos im Internet.
- Es wird noch aktiv weiterentwickelt.

# Buch



- Ich habe für diesen Kurs ein Buch geschrieben: *Programming with Python*<sup>80</sup>.
- Das Buch ist in Englisch geschrieben.
- Dieses Buch steht kostenlos im Internet.
- Es wird noch aktiv weiterentwickelt.
- **Sie** können mir helfen, es weiterzuentwickeln, durch Fragen, Anregungen, das Finden von Fehlern, usw.

# Buch



- Ich habe für diesen Kurs ein Buch geschrieben: *Programming with Python*<sup>80</sup>.
- Das Buch ist in Englisch geschrieben.
- Dieses Buch steht kostenlos im Internet.
- Es wird noch aktiv weiterentwickelt.
- **Sie** können mir helfen, es weiterzuentwickeln, durch Fragen, Anregungen, das Finden von Fehlern, usw.

URL: <https://thomasweise.github.io/programmingWithPython/programmingWithPython.pdf>

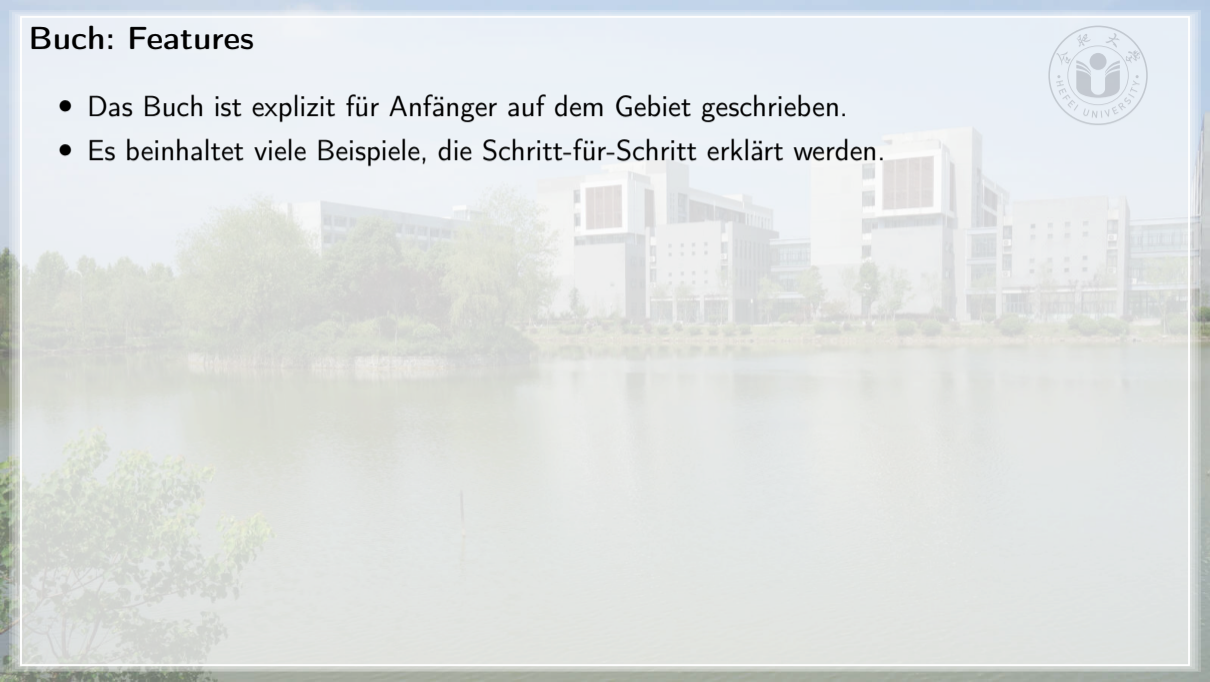
## Buch: Features

- Das Buch ist explizit für Anfänger auf dem Gebiet geschrieben.



## Buch: Features

- Das Buch ist explizit für Anfänger auf dem Gebiet geschrieben.
- Es beinhaltet viele Beispiele, die Schritt-für-Schritt erklärt werden.



## Buch: Features



- Das Buch ist explizit für Anfänger auf dem Gebiet geschrieben.
- Es beinhaltet viele Beispiele, die Schritt-für-Schritt erklärt werden.
- Fachbegriffe, die Sie vielleicht nicht kennen, werden in einem Glossary erklärt.

## Buch: Features



- Das Buch ist explizit für Anfänger auf dem Gebiet geschrieben.
- Es beinhaltet viele Beispiele, die Schritt-für-Schritt erklärt werden.
- Fachbegriffe, die Sie vielleicht nicht kennen, werden in einem Glossary erklärt.
- Das Glossary beinhaltet auch Abkürzungen, Programme, mathematische Notationen, usw.

## Buch: Features



- Das Buch ist explizit für Anfänger auf dem Gebiet geschrieben.
- Es beinhaltet viele Beispiele, die Schritt-für-Schritt erklärt werden.
- Fachbegriffe, die Sie vielleicht nicht kennen, werden in einem Glossary erklärt.
- Das Glossary beinhaltet auch Abkürzungen, Programme, mathematische Notationen, usw.
- Zum Beispiel `stdin`, UCS, UML, HTML, JSON, 1NF, breakpoint, venv, client-server architecture,  $\mathbb{R}$ ,  $\mathcal{NP}$ -hard, psql, PyPI, Matplotlib. . .

## Buch: Features



- Das Buch ist explizit für Anfänger auf dem Gebiet geschrieben.
- Es beinhaltet viele Beispiele, die Schritt-für-Schritt erklärt werden.
- Fachbegriffe, die Sie vielleicht nicht kennen, werden in einem Glossary erklärt.
- Das Glossary beinhaltet auch Abkürzungen, Programme, mathematische Notationen, usw.
- Zum Beispiel `stdin`, UCS, UML, HTML, JSON, 1NF, breakpoint, `venv`, client-server architecture,  $\mathbb{R}$ ,  $\mathcal{NP}$ -hard, `psql`, `PyPI`, `Matplotlib`...
- Diese Terme sind jeweils im Text auf das Glossary verlinkt.

## Buch: Features



- Das Buch ist explizit für Anfänger auf dem Gebiet geschrieben.
- Es beinhaltet viele Beispiele, die Schritt-für-Schritt erklärt werden.
- Fachbegriffe, die Sie vielleicht nicht kennen, werden in einem Glossary erklärt.
- Das Glossary beinhaltet auch Abkürzungen, Programme, mathematische Notationen, usw.
- Zum Beispiel `stdin`, UCS, UML, HTML, JSON, 1NF, breakpoint, venv, client-server architecture,  $\mathbb{R}$ ,  $\mathcal{NP}$ -hard, psql, PyPI, Matplotlib. . .
- Diese Terme sind jeweils im Text auf das Glossary verlinkt.
- Und im Glossary gibt es dann immer weiterführende Literaturhinweise.

# Slides und Übungen/Fragen



- Die Slides zu diesem Kurs sind ebenfalls auf der Kurswebseite online.

# Slides und Übungen/Fragen



- Die Slides zu diesem Kurs sind ebenfalls auf der Kurswebseite online.
- Ebenso finden Sie eine Liste mit Übungsfragen auf der Kurswebseite.

# Slides und Übungen/Fragen



- Die Slides zu diesem Kurs sind ebenfalls auf der Kurswebseite online.
- Ebenso finden Sie eine Liste mit Übungsfragen auf der Kurswebseite.
- URL: <https://thomasweise.github.io/programmingWithPython>

# Beispiele



- In diesem Kurs werden wir sehr viele Beispiele verwenden.

```
1 print("Hello World!")
```

```
↓ python3 very_first_program.py ↓
```

```
1 Hello World!
```

# Beispiele



- In diesem Kurs werden wir sehr viele Beispiele verwenden.
- Sie können (und sollen!) also jeden Schritt jedes Beispiels genau nachvollziehen.

```
1 print("Hello World!")
```

```
↓ python3 very_first_program.py ↓
```

```
1 Hello World!
```

# Beispiele



- In diesem Kurs werden wir sehr viele Beispiele verwenden.
- Sie können (und sollen!) also jeden Schritt jedes Beispiels genau nachvollziehen.
- Im Buch finden Sie jeweils im Titel des Programms gleich noch den Link zum Sourcecode auf GitHub.

```
1 print("Hello World!")
```

```
↓ python3 very_first_program.py ↓
```

```
1 Hello World!
```

# Komplexere Beispielprogramme



- Für einige Beispiele mit komplexeren Kommandos werden der Programmcode, die Kommandozeile zu dessen Ausführung, der standard output stream (stdout)<sup>36</sup>, der Exit-Kode<sup>34</sup>, und die Softwareversion mit angezeigt.

Link zur Datei auf Github

Listing 4.15: Listing 4.11, but with the variable explicitly hinted as `int`. (stored in file `variable_types_wrong_hints_1.py`; output in Listing 4.16)

```
1 int_var: int = 1 + 7 # Create a variable hinted as integer.
2 print(type(int_var)) # This prints "<class 'int'>".
3
4 int_var = int_var / 3 # / returns a float, but we may expect an int?
5 print(type(int_var)) # This now prints "<class 'float'>".
```

↓ `python3 variable_types_wrong_hints_1.py` ↓

Listing 4.16: The stdout of the program `variable_types_wrong_hints_1.py` given in Listing 4.15.

```
1 <class 'int'>
2 <class 'float'>
```

Kommando im Terminal

Listing 4.19: The results of static type checking the program `variable_types_wrong_hints_1.py` with Mypy of the program given in Listing 4.15.

```
1 $ mypy variable_types_wrong_hints_1.py --no-strict-optional --check-untyped
   ↳ -defs
2 variable_types_wrong_hints_1.py:4: error: Incompatible types in assignment
   ↳ (expression has type "float", variable has type "int") [assignment]
3 Found 1 error in 1 file (checked 1 source file)
4 # mypy 1.16.1 failed with exit code 1.
```

Softwareversion, Ergebnis, und Exit Code

# Komplexere Beispielprogramme



- Für einige Beispiele mit komplexeren Kommandos werden der Programmcode, die Kommandozeile zu dessen Ausführung, der standard output stream (stdout)<sup>36</sup>, der Exit-Kode<sup>34</sup>, und die Softwareversion mit angezeigt.
- Natürlich mit einem Link zum Code auf GitHub.

Link zur Datei auf Github

Listing 4.15: Listing 4.11, but `int` the variable explicitly hinted as `int`. (stored in file `variable_types_wrong_hints_1.py`; output in Listing 4.16)

```
1 int_var: int = 1 + 7 # Create a variable hinted as integer.
2 print(type(int_var)) # This prints "<class 'int'>".
3
4 int_var = int_var / 3 # / returns a float, but we may expect an int?
5 print(type(int_var)) # This now prints "<class 'float'>".
```

↓ python3 variable\_types\_wrong\_hints\_1.py ↓

Listing 4.16: The stdout of the program `variable_types_wrong_hints_1.py` given in Listing 4.15.

```
1 <class 'int'>
2 <class 'float'>
```

Kommando im Terminal

Listing 4.19: The results of static type checking the program `variable_types_wrong_hints_1.py` with Mypy of the program `mypy` given in Listing 4.15.

```
1 $ mypy variable_types_wrong_hints_1.py --no-strict-optional --check-untyped
   ↳ -defs
2 variable_types_wrong_hints_1.py:4: error: Incompatible types in assignment
   ↳ (expression has type "float", variable has type "int") [assignment]
3 Found 1 error in 1 file (checked 1 source file)
4 # mypy 1.16.1 failed with exit code 1.
```

Softwareversion, Ergebnis, und Exit Code

# Software



- Wir nutzen viele verschiedene Werkzeuge.

# Software



- Wir nutzen viele verschiedene Werkzeuge.
- Diese sind alle kostenlos im Internet verfügbar.



# Software



- Wir nutzen viele verschiedene Werkzeuge.
- Diese sind alle kostenlos im Internet verfügbar.
- Die meisten sind Open Source software.

# Software



- Wir nutzen viele verschiedene Werkzeuge.
- Diese sind alle kostenlos im Internet verfügbar.
- Die meisten sind Open Source software.
- Für jedes Werkzeug haben wir Installationshinweise ausgearbeitet.

# Software



- Wir nutzen viele verschiedene Werkzeuge.
- Diese sind alle kostenlos im Internet verfügbar.
- Die meisten sind Open Source software.
- Für jedes Werkzeug haben wir Installationshinweise ausgearbeitet. Mit Screenshots für Microsoft Windows<sup>2,25,61,68,76</sup> und Ubuntu Linux<sup>13,26</sup>.

# Literaturhinweise



- Haben Sie die kleinen roten Zahlen in dem Satz „*Mit Screenshots für Microsoft Windows<sup>2,25,61,68,76</sup> und Ubuntu Linux<sup>13,26</sup>.*“ bemerkt?

# Literaturhinweise



- Haben Sie die kleinen roten Zahlen in dem Satz „*Mit Screenshots für Microsoft Windows<sup>2,25,61,68,76</sup> und Ubuntu Linux<sup>13,26</sup>.*“ bemerkt?
- Dies sind Literaturhinweise, die am Ende der Slides gelistet werden.

# Literaturhinweise



- Haben Sie die kleinen roten Zahlen in dem Satz „*Mit Screenshots für Microsoft Windows<sup>2,25,61,68,76</sup> und Ubuntu Linux<sup>13,26</sup>.*“ bemerkt?
- Dies sind Literaturhinweise, die am Ende der Slides gelistet werden.
- Im Buch *Programming with Python<sup>80</sup>* finden Sie ebenfalls viele Literaturhinweise, dort allerdings in eckigen Klammern, also eher so: [80].

# Literaturhinweise



- Haben Sie die kleinen roten Zahlen in dem Satz „*Mit Screenshots für Microsoft Windows<sup>2,25,61,68,76</sup> und Ubuntu Linux<sup>13,26</sup>.*“ bemerkt?
- Dies sind Literaturhinweise, die am Ende der Slides gelistet werden.
- Im Buch *Programming with Python<sup>80</sup>* finden Sie ebenfalls viele Literaturhinweise, dort allerdings in eckigen Klammern, also eher so: [80].
- Wir haben für alles was wir lehren, für jedes Werkzeug, das verwendet wird, für alle Befehle und Datenformate, die wir benutzen, Literaturhinweise herausgesucht.

# Literaturhinweise



- Haben Sie die kleinen roten Zahlen in dem Satz „*Mit Screenshots für Microsoft Windows<sup>2,25,61,68,76</sup> und Ubuntu Linux<sup>13,26</sup>.*“ bemerkt?
- Dies sind Literaturhinweise, die am Ende der Slides gelistet werden.
- Im Buch *Programming with Python<sup>80</sup>* finden Sie ebenfalls viele Literaturhinweise, dort allerdings in eckigen Klammern, also eher so: [80].
- Wir haben für alles was wir lehren, für jedes Werkzeug, das verwendet wird, für alle Befehle und Datenformate, die wir benutzen, Literaturhinweise herausgesucht.
- Wir beziehen uns auf Standards, wissenschaftliche Veröffentlichungen, Dokumentationen, Vorlesungen an anderen Universitäten, und selbst auf aktuelle Gesetzgebung.

# Literaturhinweise



- Haben Sie die kleinen roten Zahlen in dem Satz „*Mit Screenshots für Microsoft Windows<sup>2,25,61,68,76</sup> und Ubuntu Linux<sup>13,26</sup>.*“ bemerkt?
- Dies sind Literaturhinweise, die am Ende der Slides gelistet werden.
- Im Buch *Programming with Python<sup>80</sup>* finden Sie ebenfalls viele Literaturhinweise, dort allerdings in eckigen Klammern, also eher so: [80].
- Wir haben für alles was wir lehren, für jedes Werkzeug, das verwendet wird, für alle Befehle und Datenformate, die wir benutzen, Literaturhinweise herausgesucht.
- Wir beziehen uns auf Standards, wissenschaftliche Veröffentlichungen, Dokumentationen, Vorlesungen an anderen Universitäten, und selbst auf aktuelle Gesetzgebung.
- Wo immer möglich, werden die entsprechenden Quellen im Literaturverzeichnis verlinkt und ich verwende, wo immer möglich, Quellen, die online verfügbar sind.

# Literaturhinweise



- Haben Sie die kleinen roten Zahlen in dem Satz „*Mit Screenshots für Microsoft Windows<sup>2,25,61,68,76</sup> und Ubuntu Linux<sup>13,26</sup>.*“ bemerkt?
- Dies sind Literaturhinweise, die am Ende der Slides gelistet werden.
- Im Buch *Programming with Python<sup>80</sup>* finden Sie ebenfalls viele Literaturhinweise, dort allerdings in eckigen Klammern, also eher so: [80].
- Wir haben für alles was wir lehren, für jedes Werkzeug, das verwendet wird, für alle Befehle und Datenformate, die wir benutzen, Literaturhinweise herausgesucht.
- Wir beziehen uns auf Standards, wissenschaftliche Veröffentlichungen, Dokumentationen, Vorlesungen an anderen Universitäten, und selbst auf aktuelle Gesetzgebung.
- Wo immer möglich, werden die entsprechenden Quellen im Literaturverzeichnis verlinkt und ich verwende, wo immer möglich, Quellen, die online verfügbar sind.
- Am Ende der Slides finden Sie oft ein Glossary (allerdings in Englisch).

# Literaturhinweise



- Haben Sie die kleinen roten Zahlen in dem Satz „*Mit Screenshots für Microsoft Windows<sup>2,25,61,68,76</sup> und Ubuntu Linux<sup>13,26</sup>.*“ bemerkt?
- Dies sind Literaturhinweise, die am Ende der Slides gelistet werden.
- Im Buch *Programming with Python<sup>80</sup>* finden Sie ebenfalls viele Literaturhinweise, dort allerdings in eckigen Klammern, also eher so: [80].
- Wir haben für alles was wir lehren, für jedes Werkzeug, das verwendet wird, für alle Befehle und Datenformate, die wir benutzen, Literaturhinweise herausgesucht.
- Wir beziehen uns auf Standards, wissenschaftliche Veröffentlichungen, Dokumentationen, Vorlesungen an anderen Universitäten, und selbst auf aktuelle Gesetzgebung.
- Wo immer möglich, werden die entsprechenden Quellen im Literaturverzeichnis verlinkt und ich verwende, wo immer möglich, Quellen, die online verfügbar sind.
- Am Ende der Slides finden Sie oft ein Glossary (allerdings in Englisch).
- Sie können *alles* nachlesen.

Alles ist da.



- Wir wissen, dass es nicht einfach ist, neue Dinge in einer fremden Sprache zu lernen.



Alles ist da.



- Wir wissen, dass es nicht einfach ist, neue Dinge in einer fremden Sprache zu lernen.
- Daher versuchen wir, alle Informationen so klar wie möglich darzustellen.

Alles ist da.



- Wir wissen, dass es nicht einfach ist, neue Dinge in einer fremden Sprache zu lernen.
- Daher versuchen wir, alle Informationen so klar wie möglich darzustellen.
- Wir stellen umfangreiches Lehrmaterial zur Verfügung.

Alles ist da.



- Wir wissen, dass es nicht einfach ist, neue Dinge in einer fremden Sprache zu lernen.
- Daher versuchen wir, alle Informationen so klar wie möglich darzustellen.
- Wir stellen umfangreiches Lehrmaterial zur Verfügung.
- Und wir machen den Kurs so praktisch wie möglich.



# Ökosystem



# Ökosystem



- Wir versuchen, ein Ökosystem von Lehrmaterial aufzubauen.

# Ökosystem



- Wir versuchen, ein Ökosystem von Lehrmaterial aufzubauen.
- Aktuell gibt es folgende beiden Kurse

# Ökosystem



- Wir versuchen, ein Ökosystem von Lehrmaterial aufzubauen.
- Aktuell gibt es folgende beiden Kurse:
  - Python, mit dem Textbuch *Thomas Weise (汤卫思). Programming with Python. Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), 2024–2026. URL:*  
*<https://thomasweise.github.io/programmingWithPython>*

# Ökosystem



- Wir versuchen, ein Ökosystem von Lehrmaterial aufzubauen.
- Aktuell gibt es folgende beiden Kurse:
  - Python, mit dem Textbuch *Thomas Weise (汤卫思). Programming with Python. Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), 2024–2026. URL: <https://thomasweise.github.io/programmingWithPython> und*
  - Datenbanken, mit dem Textbuch *Thomas Weise (汤卫思). Databases. Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), 2025. URL: <https://thomasweise.github.io/databases>.*



- Wir versuchen, ein Ökosystem von Lehrmaterial aufzubauen.
- Aktuell gibt es folgende beiden Kurse:
  - Python, mit dem Textbuch *Thomas Weise (汤卫思). Programming with Python. Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), 2024–2026. URL: <https://thomasweise.github.io/programmingWithPython> und*
  - Datenbanken, mit dem Textbuch *Thomas Weise (汤卫思). Databases. Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), 2025. URL: <https://thomasweise.github.io/databases>.*
- Diese Kurse sind miteinander verzahnt und vom Material her gleich strukturiert.



- Wir versuchen, ein Ökosystem von Lehrmaterial aufzubauen.
- Aktuell gibt es folgende beiden Kurse:
  - Python, mit dem Textbuch *Thomas Weise (汤卫思). Programming with Python. Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), 2024–2026. URL: <https://thomasweise.github.io/programmingWithPython> und*
  - Datenbanken, mit dem Textbuch *Thomas Weise (汤卫思). Databases. Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), 2025. URL: <https://thomasweise.github.io/databases>.*
- Diese Kurse sind miteinander verzahnt und vom Material her gleich strukturiert.
- Sie sind eingeladen, sich die Materialien beider Kurse anzuschauen.



- Wir versuchen, ein Ökosystem von Lehrmaterial aufzubauen.
- Aktuell gibt es folgende beiden Kurse:
  - Python, mit dem Textbuch *Thomas Weise (汤卫思). Programming with Python. Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), 2024–2026. URL: <https://thomasweise.github.io/programmingWithPython> und*
  - Datenbanken, mit dem Textbuch *Thomas Weise (汤卫思). Databases. Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), 2025. URL: <https://thomasweise.github.io/databases>.*
- Diese Kurse sind miteinander verzahnt und vom Material her gleich strukturiert.
- Sie sind eingeladen, sich die Materialien beider Kurse anzuschauen.
- Beide werden aktiv weiterentwickelt.



# Voraussetzungen



# Voraussetzungen

- Bringen Sie Ihren Laptop mit.



# Voraussetzungen



- Bringen Sie Ihren Laptop mit.
  1. Ihren eigenen Laptop.



# Voraussetzungen



- Bringen Sie Ihren Laptop mit.
  1. Ihren eigenen Laptop.
  2. Immer.



# Voraussetzungen



- Bringen Sie Ihren Laptop mit.
  1. Ihren eigenen Laptop.
  2. Immer.
  3. Vollziehen Sie unsere Beispiele nach.

# Voraussetzungen



- Bringen Sie Ihren Laptop mit.
  1. Ihren eigenen Laptop.
  2. Immer.
  3. Vollziehen Sie unsere Beispiele nach.
  4. Am Besten während der Vorlesung.

# Voraussetzungen



- Bringen Sie Ihren Laptop mit.
  1. Ihren eigenen Laptop.
  2. Immer.
  3. Vollziehen Sie unsere Beispiele nach.
  4. Am Besten während der Vorlesung.
  5. Auf Ihrem eigenen Laptop.

# Voraussetzungen



- Bringen Sie Ihren Laptop mit.
  1. Ihren eigenen Laptop.
  2. Immer.
  3. Vollziehen Sie unsere Beispiele nach.
  4. Am Besten während der Vorlesung.
  5. Auf Ihrem eigenen Laptop.
- Das sind schon alle Voraussetzungen.

# Voraussetzungen



- Bringen Sie Ihren Laptop mit.
  1. Ihren eigenen Laptop.
  2. Immer.
  3. Vollziehen Sie unsere Beispiele nach.
  4. Am Besten während der Vorlesung.
  5. Auf Ihrem eigenen Laptop.
- Das sind schon alle Voraussetzungen.
- Bonus: Im Idealfall arbeiten Sie mit dem Betriebssystem Ubuntu Linux<sup>13,26</sup>.

# Voraussetzungen



- Bringen Sie Ihren Laptop mit.
  1. Ihren eigenen Laptop.
  2. Immer.
  3. Vollziehen Sie unsere Beispiele nach.
  4. Am Besten während der Vorlesung.
  5. Auf Ihrem eigenen Laptop.
- Das sind schon alle Voraussetzungen.
- Bonus: Im Idealfall arbeiten Sie mit dem Betriebssystem Ubuntu Linux<sup>13,26</sup>. Sie könnten es ja in einer virtuellen Maschine installieren.



Warnung



# Warnung

- Sie sind chinesische Studierende.



# Warnung

- Sie sind chinesische Studierende.
- Es ist normal, zu denken: „OK, mir das auf Deutsch anzuhören oder auf Englisch zu lesen ist anstrengend. Ich lese einfach ein chinesisches Buch.“



# Warnung

- Sie sind chinesische Studierende.
- Es ist normal, zu denken: „OK, mir das auf Deutsch anzuhören oder auf Englisch zu lesen ist anstrengend. Ich lese einfach ein chinesisches Buch.“
- Ich werde ihnen auch weiterführende englische/deutsche Literatur nennen.



# Warnung

- Sie sind chinesische Studierende.
- Es ist normal, zu denken: „OK, mir das auf Deutsch anzuhören oder auf Englisch zu lesen ist anstrengend. Ich lese einfach ein chinesisches Buch.“
- Ich werde ihnen auch weiterführende englische/deutsche Literatur nennen.
- **ABER: Lesen Sie andere Bücher nur *begleitend*, um Aspekte zu klären, die Sie vielleicht in meinem Material nicht verstanden haben!**



# Warnung

- Sie sind chinesische Studierende.
- Es ist normal, zu denken: „OK, mir das auf Deutsch anzuhören oder auf Englisch zu lesen ist anstrengend. Ich lese einfach ein chinesisches Buch.“
- Ich werde ihnen auch weiterführende englische/deutsche Literatur nennen.
- **ABER: Lesen Sie andere Bücher nur *begleitend*, um Aspekte zu klären, die Sie vielleicht in meinem Material nicht verstanden haben!**
- Sie **müssen** mein Material zu diesem Kurs lesen, wenn Sie bestehen wollen.



# Warnung



- Sie sind chinesische Studierende.
- Es ist normal, zu denken: „OK, mir das auf Deutsch anzuhören oder auf Englisch zu lesen ist anstrengend. Ich lese einfach ein chinesisches Buch.“
- Ich werde ihnen auch weiterführende englische/deutsche Literatur nennen.
- **ABER: Lesen Sie andere Bücher nur *begleitend*, um Aspekte zu klären, die Sie vielleicht in meinem Material nicht verstanden haben!**
- Sie **müssen** mein Material zu diesem Kurs lesen, wenn Sie bestehen wollen.
- Dafür gibt es viele Gründe

# Warnung



- Ich werde ihnen auch weiterführende englische/deutsche Literatur nennen.
- **ABER:** Lesen Sie andere Bücher nur *begleitend*, um Aspekte zu klären, die Sie vielleicht in meinem Material nicht verstanden haben!
- Sie **müssen** mein Material zu diesem Kurs lesen, wenn Sie bestehen wollen.
- Dafür gibt es viele Gründe
  1. Viele existierende Bücher sind veraltet oder haben einen anderen Fokus.

# Warnung



- Ich werde ihnen auch weiterführende englische/deutsche Literatur nennen.
- **ABER:** Lesen Sie andere Bücher nur *begleitend*, um Aspekte zu klären, die Sie vielleicht in meinem Material nicht verstanden haben!
- Sie **müssen** mein Material zu diesem Kurs lesen, wenn Sie bestehen wollen.
- Dafür gibt es viele Gründe
  1. Viele existierende Bücher sind veraltet oder haben einen anderen Fokus.
  2. Wir lernen die *aktuellen und modernsten* Aspekte der praktischen Arbeit auf unserem Gebiet kennen.

# Warnung



- Ich werde ihnen auch weiterführende englische/deutsche Literatur nennen.
- **ABER:** Lesen Sie andere Bücher nur *begleitend*, um Aspekte zu klären, die Sie vielleicht in meinem Material nicht verstanden haben!
- Sie **müssen** mein Material zu diesem Kurs lesen, wenn Sie bestehen wollen.
- Dafür gibt es viele Gründe
  1. Viele existierende Bücher sind veraltet oder haben einen anderen Fokus.
  2. Wir lernen die *aktuellen und modernsten* Aspekte der praktischen Arbeit auf unserem Gebiet kennen.
  3. Sehr viele unserer Themen kommen **NICHT** in anderen Büchern vor!

# Warnung



- Ich werde ihnen auch weiterführende englische/deutsche Literatur nennen.
- **ABER:** Lesen Sie andere Bücher nur *begleitend*, um Aspekte zu klären, die Sie vielleicht in meinem Material nicht verstanden haben!
- Sie **müssen** mein Material zu diesem Kurs lesen, wenn Sie bestehen wollen.
- Dafür gibt es viele Gründe
  1. Viele existierende Bücher sind veraltet oder haben einen anderen Fokus.
  2. Wir lernen die *aktuellen und modernsten* Aspekte der praktischen Arbeit auf unserem Gebiet kennen.
  3. Sehr viele unserer Themen kommen **NICHT** in anderen Büchern vor!
  4. Sie kommen aber alle in der Prüfung vor!

# Warnung



- Ich werde ihnen auch weiterführende englische/deutsche Literatur nennen.
- **ABER:** Lesen Sie andere Bücher nur *begleitend*, um Aspekte zu klären, die Sie vielleicht in meinem Material nicht verstanden haben!
- Sie **müssen** mein Material zu diesem Kurs lesen, wenn Sie bestehen wollen.
- Dafür gibt es viele Gründe
  1. Viele existierende Bücher sind veraltet oder haben einen anderen Fokus.
  2. Wir lernen die *aktuellen und modernsten* Aspekte der praktischen Arbeit auf unserem Gebiet kennen.
  3. Sehr viele unserer Themen kommen **NICHT** in anderen Büchern vor!
  4. Sie kommen aber alle in der Prüfung vor!
  5. Wenn Sie nicht unser Material durcharbeiten, dann sitzen Sie nachher in einer Prüfung und verstehen nicht mal die Fragen. . .

# Warnung



- **ABER:** Lesen Sie andere Bücher nur *begleitend*, um Aspekte zu klären, die Sie vielleicht in meinem Material nicht verstanden haben!
- Sie **müssen** mein Material zu diesem Kurs lesen, wenn Sie bestehen wollen.
- Dafür gibt es viele Gründe
  1. Viele existierende Bücher sind veraltet oder haben einen anderen Fokus.
  2. Wir lernen die *aktuellen und modernsten* Aspekte der praktischen Arbeit auf unserem Gebiet kennen.
  3. Sehr viele unserer Themen kommen **NICHT** in anderen Büchern vor!
  4. Sie kommen aber alle in der Prüfung vor!
  5. Wenn Sie nicht unser Material durcharbeiten, dann sitzen Sie nachher in einer Prüfung und verstehen nicht mal die Fragen. . .
  6. Wenn Sie sich das Wissen nur mit anderen Materialien aneignen wollen, **dann werden Sie zu 100% durch die Prüfung durchfallen!**

# Warnung



- **ABER:** Lesen Sie andere Bücher nur *begleitend*, um Aspekte zu klären, die Sie vielleicht in meinem Material nicht verstanden haben!
- Sie **müssen** mein Material zu diesem Kurs lesen, wenn Sie bestehen wollen.
- Dafür gibt es viele Gründe
  1. Viele existierende Bücher sind veraltet oder haben einen anderen Fokus.
  2. Wir lernen die *aktuellen und modernsten* Aspekte der praktischen Arbeit auf unserem Gebiet kennen.
  3. Sehr viele unserer Themen kommen **NICHT** in anderen Büchern vor!
  4. Sie kommen aber alle in der Prüfung vor!
  5. Wenn Sie nicht unser Material durcharbeiten, dann sitzen Sie nachher in einer Prüfung und verstehen nicht mal die Fragen. . .
  6. Wenn Sie sich das Wissen nur mit anderen Materialien aneignen wollen, **dann werden Sie zu 100% durch die Prüfung durchfallen!**
- **Hören Sie in der Vorlesung zu! Lesen Sie mein Material (englisch oder deutsch)!**



# Zusammenfassung



# Kurs Programmieren mit Python



- Willkommen zum Kurs **Programmieren mit Python**.



# Kurs Programmieren mit Python



- Willkommen zum Kurs **Programmieren mit Python**.
- Ich freue mich auf unsere Zusammenarbeit und hoffe, dass wir Spaß mit diesem Gebiet haben werden.

# Kurs Programmieren mit Python



- Willkommen zum Kurs **Programmieren mit Python**.
- Ich freue mich auf unsere Zusammenarbeit und hoffe, dass wir Spaß mit diesem Gebiet haben werden.
- Und: Bitte besuchen Sie die Vorlesung, hören Sie zu, und verwenden/lesen Sie mein Vorlesungsmaterial. . .



谢谢你们！  
Thank you!  
Vielen Dank!



# References I



- [1] David J. Agans. *Debugging*. New York, NY, USA: AMACOM, Sep. 2002. ISBN: 978-0-8144-2678-4 (siehe S. 151).
- [2] Daniel J. Barrett. *Efficient Linux at the Command Line*. Sebastopol, CA, USA: O'Reilly Media, Inc., Feb. 2022. ISBN: 978-1-0981-1340-7 (siehe S. 88–100, 152, 154).
- [3] Tim Berners-Lee. *Re: Qualifiers on Hypertext links...* Geneva, Switzerland: World Wide Web project, European Organization for Nuclear Research (CERN) und Newsgroups: alt.hypertext, 6. Aug. 1991. URL: <https://www.w3.org/People/Berners-Lee/1991/08/art-6484.txt> (besucht am 2025-02-05) (siehe S. 152, 155).
- [4] Alex Berson. *Client/Server Architecture*. 2. Aufl. Computer Communications Series. New York, NY, USA: McGraw-Hill, 29. März 1996. ISBN: 978-0-07-005664-0 (siehe S. 151).
- [5] Ethan Bommarito und Michael Bommarito. *An Empirical Analysis of the Python Package Index (PyPI)*. arXiv.org: Computing Research Repository (CoRR) abs/1907.11073. Ithaca, NY, USA: Cornell University Library, 26. Juli 2019. doi:10.48550/arXiv.1907.11073. URL: <https://arxiv.org/abs/1907.11073> (besucht am 2024-08-17). arXiv:1907.11073v2 [cs.SE] 26 Jul 2019 (siehe S. 153).
- [6] Grady Booch, James Rumbaugh und Ivar Jacobson. *The Unified Modeling Language Reference Manual*. 1. Aufl. Reading, MA, USA: Addison-Wesley Professional, Jan. 1999. ISBN: 978-0-201-57168-4 (siehe S. 154).
- [7] Ed Bott. *Windows 11 Inside Out*. Hoboken, NJ, USA: Microsoft Press, Pearson Education, Inc., Feb. 2023. ISBN: 978-0-13-769132-6 (siehe S. 152).
- [8] Ron Brash und Ganesh Naik. *Bash Cookbook*. Birmingham, England, UK: Packt Publishing Ltd, Juli 2018. ISBN: 978-1-78862-936-2 (siehe S. 151).
- [9] Tim Bray. *The JavaScript Object Notation (JSON) Data Interchange Format*. Request for Comments (RFC) 8259. Wilmington, DE, USA: Internet Engineering Task Force (IETF), Dez. 2017. URL: <https://www.ietf.org/rfc/rfc8259.txt> (besucht am 2025-02-05) (siehe S. 152).
- [10] Jason Cannon. *High Availability for the LAMP Stack*. Shelter Island, NY, USA: Manning Publications, Juni 2022 (siehe S. 153).
- [11] Josh Centers. *Take Control of iOS 18 and iPadOS 18*. San Diego, CA, USA: Take Control Books, Dez. 2024. ISBN: 978-1-990783-55-5 (siehe S. 152).

# References II



- [12] Bo Chen, Chris N. Potts und Gerhard J. Woeginger. "A Review of Machine Scheduling: Complexity, Algorithms and Approximability". In: *Handbook of Combinatorial Optimization*. Hrsg. von Panos Miltiades Pardalos, Ding-Zhu Du und Ronald Lewis Graham. 1. Aufl. Boston, MA, USA: Springer, 1998, S. 1493–1641. ISBN: 978-1-4613-7987-4. doi:10.1007/978-1-4613-0303-9\_25. See also pages 21–169 in volume 3/3 by Norwell, MA, USA: Kluwer Academic Publishers. (Siehe S. 155).
- [13] David Clinton und Christopher Negus. *Ubuntu Linux Bible*. 10. Aufl. Bible Series. Chichester, West Sussex, England, UK: John Wiley and Sons Ltd., 10. Nov. 2020. ISBN: 978-1-119-72233-5 (siehe S. 88–100, 114–122, 154).
- [14] Edgar Frank „Ted“ Codd. *Normalized Data Base Structure: A Brief Tutorial*. IBM Research Report RJ935. San Jose, CA, USA: IBM Research Laboratory, 1971. URL: [https://www.fsmwarden.com/Codd/Normalized%20data%20base%20structure\\_%20a%20brief%20tutorial\(1971,%20nov\).pdf](https://www.fsmwarden.com/Codd/Normalized%20data%20base%20structure_%20a%20brief%20tutorial(1971,%20nov).pdf) (besucht am 2025-05-04) (siehe S. 143).
- [15] Edgar Frank „Ted“ Codd. "Normalized Data Base Structure: A Brief Tutorial". In: *ACM SIGFIDET Workshop on Data Description, Access, and Control*. 11.–12. Nov. 1971, San Diego, CA, USA. Hrsg. von Edgar Frank „Ted“ Codd und Albert L. Dean Jr. New York, NY, USA: Association for Computing Machinery (ACM), 1971, S. 1–17. ISBN: 978-1-4503-7300-5. doi:10.1145/1734714.1734716. See also<sup>14</sup> (siehe S. 151).
- [16] Edgar Frank „Ted“ Codd. "A Relational Model of Data for Large Shared Data Banks". *Communications of the ACM (CACM)* 13(6):377–387, Juni 1970. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 0001-0782. doi:10.1145/362384.362685. URL: <https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf> (besucht am 2025-01-05) (siehe S. 151).
- [17] Edgar Frank „Ted“ Codd. *Further Normalization of the Data Base Relational Model*. IBM Research Report RJ909. San Jose, CA, USA: IBM Research Laboratory, 31. Aug. 1971. URL: <https://forum.thethirdmanifesto.com/wp-content/uploads/asgarosforum/987737/00-efc-further-normalization.pdf> (besucht am 2025-05-04). Reprinted in and presented at<sup>56</sup> (siehe S. 151).
- [18] Stephen Arthur Cook. "The Complexity of Theorem-Proving Procedures". In: *Third Annual ACM Symposium on Theory of Computing (STOC'1971)*. 3.–5. Mai 1971, Shaker Heights, OH, USA. Hrsg. von Michael A. Harrison, Ranan B. Banerji und Jeffrey D. Ullman. New York, NY, USA: Association for Computing Machinery (ACM), 1971, S. 151–158. ISBN: 978-1-4503-7464-4. doi:10.1145/800157.805047 (siehe S. 155).

# References III



- [19] Christopher J. Date. *An Introduction to Database Systems*. 8. Aufl. Hoboken, NJ, USA: Pearson Education, Inc., Juli 2003. ISBN: **978-0-321-19784-9** (siehe S. **151, 153**).
- [20] Paul Deitel, Harvey Deitel und Abbey Deitel. *Internet & World Wide Web: How to Program*. 5. Aufl. Hoboken, NJ, USA: Pearson Education, Inc., Nov. 2011. ISBN: **978-0-13-299045-5** (siehe S. **155**).
- [21] *ECMAScript Language Specification*. Standard ECMA-262, 3rd Edition. Geneva, Switzerland: Ecma International, Dez. 1999. URL: [https://ecma-international.org/wp-content/uploads/ECMA-262\\_3rd\\_edition\\_december\\_1999.pdf](https://ecma-international.org/wp-content/uploads/ECMA-262_3rd_edition_december_1999.pdf) (besucht am 2024-12-15) (siehe S. **152**).
- [22] Ramez Elmasri und Shamkant Navathe. *Fundamentals of Database Systems*. 7. Aufl. Hoboken, NJ, USA: Pearson Education, Inc., Juni 2015. ISBN: **978-0-13-397077-7** (siehe S. **151, 153**).
- [23] Luca Ferrari und Enrico Pirozzi. *Learn PostgreSQL*. 2. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Okt. 2023. ISBN: **978-1-83763-564-1** (siehe S. **153**).
- [24] Michael T. Goodrich. *A Gentle Introduction to NP-Completeness*. Irvine, CA, USA: University of California, Irvine, Apr. 2022. URL: <https://ics.uci.edu/~goodrich/teach/cs165/notes/NPComplete.pdf> (besucht am 2025-08-01) (siehe S. **155**).
- [25] Michael Hausenblas. *Learning Modern Linux*. Sebastopol, CA, USA: O'Reilly Media, Inc., Apr. 2022. ISBN: **978-1-0981-0894-6** (siehe S. **88–100, 152**).
- [26] Matthew Helmke. *Ubuntu Linux Unleashed 2021 Edition*. 14. Aufl. Reading, MA, USA: Addison-Wesley Professional, Aug. 2020. ISBN: **978-0-13-668539-5** (siehe S. **88–100, 114–122, 154**).
- [27] Ian Hickson, Robin Berjon, Steve Faulkner, Travis Leithead, Erika Doyle Navara, Theresa O'Connor und Silvia Pfeiffer, Hrsg. *HTML5: A Vocabulary and Associated APIs for HTML and XHTML*. W3C Recommendation. Wakefield, MA, USA: World Wide Web Consortium (W3C), 28. Okt. 2014. URL: <http://www.w3.org/TR/2014/REC-html5-20141028> (besucht am 2024-12-17) (siehe S. **152**).
- [28] John Hunt. *A Beginners Guide to Python 3 Programming*. 2. Aufl. Undergraduate Topics in Computer Science (UTICS). Cham, Switzerland: Springer, 2023. ISBN: **978-3-031-35121-1**. doi:[10.1007/978-3-031-35122-8](https://doi.org/10.1007/978-3-031-35122-8) (siehe S. **153**).
- [29] John D. Hunter. "Matplotlib: A 2D Graphics Environment". *Computing in Science & Engineering* 9(3):90–95, Mai–Juni 2007. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE). ISSN: **1521-9615**. doi:[10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55) (siehe S. **152**).

# References IV



- [30] John D. Hunter, Darren Dale, Eric Firing, Michael Droettboom und The Matplotlib Development Team. *Matplotlib: Visualization with Python*. Austin, TX, USA: NumFOCUS, Inc., 2012–2025. URL: <https://matplotlib.org> (besucht am 2025-02-02) (siehe S. 152).
- [31] *Information Technology – Universal Coded Character Set (UCS)*. International Standard ISO/IEC 10646:2020. Geneva, Switzerland: International Organization for Standardization (ISO) und International Electrotechnical Commission (IEC), Dez. 2020 (siehe S. 154).
- [32] *Python 3 Documentation. Installing Python Modules*. Beaverton, OR, USA: Python Software Foundation (PSF), 2001–2025. URL: <https://docs.python.org/3/installing> (besucht am 2024-08-17) (siehe S. 153).
- [33] Robert Johansson. *Numerical Python: Scientific Computing and Data Science Applications with NumPy, SciPy and Matplotlib*. New York, NY, USA: Apress Media, LLC, Dez. 2018. ISBN: 978-1-4842-4246-9 (siehe S. 152).
- [34] „exit – Terminate a Process“. In: *POSIX.1-2024: The Open Group Base Specifications Issue 8, IEEE Std 1003.1™-2024 Edition*. Hrsg. von Andrew Josey. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE) und San Francisco, CA, USA: The Open Group, 8. Aug. 2024. URL: <https://pubs.opengroup.org/onlinepubs/9799919799/functions/exit.html> (besucht am 2024-10-30) (siehe S. 86, 87).
- [35] Andrew Josey, Hrsg. *POSIX.1-2024: The Open Group Base Specifications Issue 8, IEEE Std 1003.1™-2024 Edition*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE) und San Francisco, CA, USA: The Open Group, 8. Aug. 2024. URL: <https://pubs.opengroup.org/onlinepubs/9799919799> (besucht am 2024-10-30).
- [36] „stderr, stdin, stdout – Standard I/O Streams“. In: *POSIX.1-2024: The Open Group Base Specifications Issue 8, IEEE Std 1003.1™-2024 Edition*. Hrsg. von Andrew Josey. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers (IEEE) und San Francisco, CA, USA: The Open Group, 8. Aug. 2024. URL: <https://pubs.opengroup.org/onlinepubs/9799919799/functions/stdin.html> (besucht am 2024-10-30) (siehe S. 86, 87, 153, 154).
- [37] William (Bill) Kent. „A Simple Guide to Five Normal Forms in Relational Database Theory“. *Communications of the ACM (CACM)* 26(2):120–125, Sep. 1982–Feb. 1983. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 0001-0782. doi:10.1145/358024.358054. URL: <https://www.cs.dartmouth.edu/~cs61/Resources/Papers/CACM%20Kent%20Five%20Normal%20Forms.pdf> (besucht am 2025-05-03) (siehe S. 151, 153).

# References V



- [38] Jay LaCroix. *Mastering Ubuntu Server*. 4. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Sep. 2022. ISBN: **978-1-80323-424-3** (siehe S. 153).
- [39] Eugene Leighton Lawler, Jan Karel Lenstra, Alexander Hendrik George Rinnooy Kan und David B. Shmoys. "Sequencing and Scheduling: Algorithms and Complexity". In: *Production Planning and Inventory*. Hrsg. von Stephen C. Graves, Alexander Hendrik George Rinnooy Kan und Paul H. Zipkin. Bd. IV der Reihe Handbooks of Operations Research and Management Science. Amsterdam, The Netherlands: Elsevier B.V., 1993. Kap. 9, S. 445–522. ISSN: **0927-0507**. ISBN: **978-0-444-87472-6**. doi:10.1016/S0927-0507(05)80189-6. URL: <http://alexandria.tue.nl/repository/books/339776.pdf> (besucht am 2023-12-06) (siehe S. 155).
- [40] Kent D. Lee und Steve Hubbard. *Data Structures and Algorithms with Python*. Undergraduate Topics in Computer Science (UTICS). Cham, Switzerland: Springer, 2015. ISBN: **978-3-319-13071-2**. doi:10.1007/978-3-319-13072-9 (siehe S. 153).
- [41] Gloria Lotha, Aakanksha Gaur, Erik Gregersen, Swati Chopra und William L. Hosch. "Client-Server Architecture". In: *Encyclopaedia Britannica*. Hrsg. von The Editors of Encyclopaedia Britannica. Chicago, IL, USA: Encyclopædia Britannica, Inc., 3. Jan. 2025. URL: <https://www.britannica.com/technology/client-server-architecture> (besucht am 2025-01-20) (siehe S. 151).
- [42] Mark Lutz. *Learning Python*. 6. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2025. ISBN: **978-1-0981-7130-8** (siehe S. 153).
- [43] Carl Meyer. *Python Virtual Environments*. Python Enhancement Proposal (PEP) 405. Beaverton, OR, USA: Python Software Foundation (PSF), 13. Juni 2011–24. Mai 2012. URL: <https://peps.python.org/pep-0405> (besucht am 2024-12-25) (siehe S. 155).
- [44] Cameron Newham und Bill Rosenblatt. *Learning the Bash Shell – Unix Shell Programming: Covers Bash 3.0*. 3. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., 2005. ISBN: **978-0-596-00965-6** (siehe S. 151).
- [45] Regina O. Obe und Leo S. Hsu. *PostgreSQL: Up and Running*. 3. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., Okt. 2017. ISBN: **978-1-4919-6336-4** (siehe S. 153).
- [46] **OMG® Unified Modeling Language® (OMG UML®)**. Version 2.5.1. OMG Document formal/2017-12-05. Milford, MA, USA: Object Management Group, Inc. (OMG), Dez. 2017. URL: <https://www.omg.org/spec/UML/2.5.1/PDF> (besucht am 2025-03-30) (siehe S. 154).

# References VI



- [47] Robert Orfali, Dan Harkey und Jeri Edwards. *Client/Server Survival Guide*. 3. Aufl. Chichester, West Sussex, England, UK: John Wiley and Sons Ltd., 25. Jan. 1999. ISBN: **978-0-471-31615-2** (siehe S. 151).
- [48] Ashwin Pajankar. *Hands-on Matplotlib: Learn Plotting and Visualizations with Python 3*. New York, NY, USA: Apress Media, LLC, Nov. 2021. ISBN: **978-1-4842-7410-1** (siehe S. 152).
- [49] Yasset Pérez-Riverol, Laurent Gatto, Rui Wang, Timo Sachsenberg, Julian Uszkoreit, Felipe da Veiga Leprevost, Christian Fufezan, Tobias Ternent, Stephen J. Eglén, Daniel S. Katz, Tom J. Pollard, Alexander Kononov, Robert M. Flight, Kai Blin und Juan Antonio Vizcaíno. "Ten Simple Rules for Taking Advantage of Git and GitHub". *PLOS Computational Biology* 12(7), 14. Juli 2016. San Francisco, CA, USA: Public Library of Science (PLOS). ISSN: **1553-7358**. doi:[10.1371/JOURNAL.PCBI.1004947](https://doi.org/10.1371/JOURNAL.PCBI.1004947) (siehe S. 152).
- [50] pip Developers. *pip Documentation v24.3.1*. Beaverton, OR, USA: Python Software Foundation (PSF), 27. Okt. 2024. URL: <https://pip.pypa.io> (besucht am 2024-12-25) (siehe S. 153).
- [51] *PostgreSQL Essentials: Leveling Up Your Data Work*. Sebastopol, CA, USA: O'Reilly Media, Inc., März 2024 (siehe S. 153).
- [52] Abhishek Ratan, Eric Chou, Pradeeban Kathiravelu und Dr. M.O. Faruque Sarker. *Python Network Programming*. Birmingham, England, UK: Packt Publishing Ltd, Jan. 2019. ISBN: **978-1-78883-546-6** (siehe S. 151).
- [53] Mark Richards und Neal Ford. *Fundamentals of Software Architecture: An Engineering Approach*. Sebastopol, CA, USA: O'Reilly Media, Inc., Jan. 2020. ISBN: **978-1-4920-4345-4** (siehe S. 151).
- [54] Kristian Rother. *Pro Python Best Practices: Debugging, Testing and Maintenance*. New York, NY, USA: Apress Media, LLC, März 2017. ISBN: **978-1-4842-2241-6** (siehe S. 151).
- [55] Ernest E. Rothman, Rich Rosen und Brian Jepson. *Mac OS X for Unix Geeks*. 4. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., Sep. 2008. ISBN: **978-0-596-52062-5** (siehe S. 152).
- [56] Randall Rustin, Hrsg. *Data Base Systems: Courant Computer Science Symposium 6*. 24.–25. Mai 1971, New York, NY, USA. Englewood Cliffs, NJ, USA: Prentice-Hall, 1972. ISBN: **978-0-13-196741-0** (siehe S. 143).
- [57] Ahmad Sahar. *iOS 26 Programming for Beginners*. 10. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Nov. 2025. ISBN: **978-1-80602-393-6** (siehe S. 155).

# References VII



- [58] Yuriy Shamshin. *Databases*. Riga, Latvia: ISMA University of Applied Sciences, Mai 2024. URL: <https://dbs.academy.lv> (besucht am 2025-01-11).
- [59] Yuriy Shamshin. "Normalization". In: *Databases*. Riga, Latvia: ISMA University of Applied Sciences, Mai 2024. Kap. 07a. URL: [https://dbs.academy.lv/lection/dbs\\_LS07ENa\\_normalization.pdf](https://dbs.academy.lv/lection/dbs_LS07ENa_normalization.pdf) (besucht am 2025-05-03) (siehe S. 153).
- [60] Yuriy Shamshin. "RDM Normalization. Data Anomalies. Functional Dependency. Normal Forms.". In: *Databases*. Riga, Latvia: ISMA University of Applied Sciences, Mai 2024. Kap. 07. URL: [https://dbs.academy.lv/lection/dbs\\_LS07EN\\_normalization.pdf](https://dbs.academy.lv/lection/dbs_LS07EN_normalization.pdf) (besucht am 2025-05-03) (siehe S. 153).
- [61] Ellen Siever, Stephen Figgins, Robert Love und Arnold Robbins. *Linux in a Nutshell*. 6. Aufl. Sebastopol, CA, USA: O'Reilly Media, Inc., Sep. 2009. ISBN: 978-0-596-15448-6 (siehe S. 88–100, 152).
- [62] Anna Skoulikari. *Learning Git*. Sebastopol, CA, USA: O'Reilly Media, Inc., Mai 2023. ISBN: 978-1-0981-3391-7 (siehe S. 152).
- [63] Drew Smith. *Modern Apple Platform Administration – macOS and iOS Essentials (2025)*. Birmingham, England, UK: Packt Publishing Ltd, Feb. 2025. ISBN: 978-1-80580-309-6 (siehe S. 152).
- [64] Alkin Tezuysal und Ibrar Ahmed. *Database Design and Modeling with PostgreSQL and MySQL*. Birmingham, England, UK: Packt Publishing Ltd, Juli 2024. ISBN: 978-1-80323-347-5 (siehe S. 153).
- [65] *The JSON Data Interchange Syntax*. Standard ECMA-404, 2nd Edition. Geneva, Switzerland: Ecma International, Dez. 2017. URL: <https://ecma-international.org/publications-and-standards/standards/ecma-404> (besucht am 2024-12-15) (siehe S. 152).
- [66] *The Python Package Index (PyPI)*. Beaverton, OR, USA: Python Software Foundation (PSF), 2024. URL: <https://pypi.org> (besucht am 2024-08-17) (siehe S. 153).
- [67] *The Unicode Standard, Version 15.1: Archived Code Charts*. South San Francisco, CA, USA: The Unicode Consortium, 25. Aug. 2023. URL: <https://www.unicode.org/Public/15.1.0/charts/CodeCharts.pdf> (besucht am 2024-07-26) (siehe S. 154).
- [68] Linus Torvalds. "The Linux Edge". *Communications of the ACM (CACM)* 42(4):38–39, Apr. 1999. New York, NY, USA: Association for Computing Machinery (ACM). ISSN: 0001-0782. doi:10.1145/299157.299165 (siehe S. 88–100, 152).

# References VIII



- [69] Brad Traversy. *Modern HTML & CSS From The Beginning 2.0*. 2. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Juli 2024. ISBN: **978-1-83588-056-2** (siehe S. 152).
- [70] Mariot Tsitoara. *Beginning Git and GitHub: Version Control, Project Management and Teamwork for the New Developer*. New York, NY, USA: Apress Media, LLC, März 2024. ISBN: **979-8-8688-0215-7** (siehe S. 152, 155).
- [71] *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 3. Aufl. The Addison-Wesley Object Technology Series. Reading, MA, USA: Addison-Wesley Professional, Sep. 2003. ISBN: **978-0-321-19368-1** (siehe S. 154).
- [72] *UML Notation Guide*. Version 1.1. Santa Clara, CA, USA: Rational Software Corporation, Redmond, WA, USA: Microsoft Corporation, Palo Alto, CA, USA: Hewlett-Packard Company, Redwood Shores, CA, USA: Oracle Corporation, Dallas, TX, USA: Sterling Software, Ottawa, ON, Canada: MCI Systemhouse Corporation, Blue Bell, PA, USA: Unisys Corporation, Blue Bell, PA, USA: ICON Computing, Santa Clara, CA, USA: IntelliCorp, Burlington, MA, USA: i-Logix, Armonk, NY, USA: International Business Machines Corporation (IBM), Kanata, ON, Canada: ObjecTime Limited, Chicago, IL, USA: Platinum Technology Inc., Boston, MA, USA: Ptech Inc., Orlando, FL, USA: Taskon A/S, Paoli, PA, USA: Reich Technologies und Paris, Île-de-France, France: Softeam, 1. Sep. 1997. URL: <https://web.cse.msu.edu/~cse870/Materials/uml-notation-guide-9-97.pdf> (besucht am 2025-03-30) (siehe S. 154).
- [73] *Unicode®15.1.0*. South San Francisco, CA, USA: The Unicode Consortium, 12. Sep. 2023. ISBN: **978-1-936213-33-7**. URL: <https://www.unicode.org/versions/Unicode15.1.0> (besucht am 2024-07-26) (siehe S. 154).
- [74] Marat Valiev, Bogdan Vasilescu und James D. Herbsleb. "Ecosystem-Level Determinants of Sustained Activity in Open-Source Projects: A Case Study of the PyPI Ecosystem". In: *ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/SIGSOFT FSE'2018)*. 4.–9. Nov. 2018, Lake Buena Vista, FL, USA. Hrsg. von Gary T. Leavens, Alessandro F. Garcia und Corina S. Păsăreanu. New York, NY, USA: Association for Computing Machinery (ACM), 2018, S. 644–655. ISBN: **978-1-4503-5573-5**. doi:10.1145/3236024.3236062 (siehe S. 153).
- [75] Bruce M. Van Horn II und Quan Nguyen. *Hands-On Application Development with PyCharm*. 2. Aufl. Birmingham, England, UK: Packt Publishing Ltd, Okt. 2023. ISBN: **978-1-83763-235-0** (siehe S. 153).
- [76] Sander van Vugt. *Linux Fundamentals*. 2. Aufl. Hoboken, NJ, USA: Pearson IT Certification, Juni 2022. ISBN: **978-0-13-792931-3** (siehe S. 88–100, 152).

# References IX



- [77] "[venv](#) – Creation of Virtual Environments". In: *Python 3 Documentation. The Python Standard Library*. Beaverton, OR, USA: Python Software Foundation (PSF), 2001–2025. URL: <https://docs.python.org/3/library/venv.html> (besucht am 2024-12-25) (siehe S. 155).
- [78] "Virtual Environments and Packages". In: *Python 3 Documentation. The Python Tutorial*. Beaverton, OR, USA: Python Software Foundation (PSF), 2001–2025. Kap. 12. URL: <https://docs.python.org/3/tutorial/venv.html> (besucht am 2024-12-24) (siehe S. 155).
- [79] Thomas Weise (汤卫思). *Databases*. Hefei, Anhui, China (中国安徽省合肥市): Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), 2025. URL: <https://thomasweise.github.io/databases> (besucht am 2025-01-05) (siehe S. 106–112, 151).
- [80] Thomas Weise (汤卫思). *Programming with Python*. Hefei, Anhui, China (中国安徽省合肥市): Hefei University (合肥大学), School of Artificial Intelligence and Big Data (人工智能与大数据学院), 2024–2026. URL: <https://thomasweise.github.io/programmingWithPython> (besucht am 2025-01-05) (siehe S. 67–72, 86, 87, 93–100, 106–112, 151, 153).
- [81] Kevin Wilson. *Python Made Easy*. Birmingham, England, UK: Packt Publishing Ltd, Aug. 2024. ISBN: 978-1-83664-615-0 (siehe S. 151, 153).
- [82] Martin Yanev. *PyCharm Productivity and Debugging Techniques*. Birmingham, England, UK: Packt Publishing Ltd, Okt. 2022. ISBN: 978-1-83763-244-2 (siehe S. 153).
- [83] Kinza Yasar und Craig S. Mullins. *Definition: Database Management System (DBMS)*. Newton, MA, USA: TechTarget, Inc., Juni 2024. URL: <https://www.techtarget.com/searchdatamanagement/definition/database-management-system> (besucht am 2025-01-11) (siehe S. 151).
- [84] Giorgio Zarrelli. *Mastering Bash*. Birmingham, England, UK: Packt Publishing Ltd, Juni 2017. ISBN: 978-1-78439-687-9 (siehe S. 151).

# Glossary (in English) I



1NF The first normal form (NF) in relational databases (DBs)<sup>16,19,22,37</sup>.

2NF The second normal form (NF) in relational DBs<sup>15,17,19,22,37</sup>.

3NF The third normal form (NF) in relational DBs<sup>15,17,19,22,37</sup>.

Bash is a shell used under Ubuntu Linux, i.e., the program that „runs“ in the terminal and interprets your commands, allowing you to start and interact with other programs<sup>8,44,84</sup>. Learn more at <https://www.gnu.org/software/bash>.

breakpoint A breakpoint is a mark in a line of code in an Integrated Development Environment (IDE) at which the debugger will pause the execution of a program.

client In a client-server architecture, the client is a device or process that requests a service from the server. It initiates the communication with the server, sends a request, and receives the response with the result of the request. Typical examples for clients are web browsers in the internet as well as clients for database management systems (DBMSes), such as psql.

client-server architecture is a system design where a central server receives requests from one or multiple clients<sup>4,41,47,52,53</sup>. These requests and responses are usually sent over network connections. A typical example for such a system is the World Wide Web (WWW), where web servers host websites and make them available to web browsers, the clients. Another typical example is the structure of DB software, where a central server, the DBMS, offers access to the DB to the different clients. Here, the client can be some terminal software shipping with the DBMS, such as psql, or the different applications that access the DBs.

DB A *database* is an organized collection of structured information or data, typically stored electronically in a computer system. Databases are discussed in our book *Databases*<sup>79</sup>.

DBMS A *database management system* is the software layer located between the user or application and the DB. The DBMS allows the user/application to create, read, write, update, delete, and otherwise manipulate the data in the DB<sup>83</sup>.

debugger A debugger is a tool that lets you execute a program step-by-step while observing the current values of variables. This allows you to find errors in the code more easily<sup>1,54,81</sup>. Learn more about debugging in<sup>80</sup>.

# Glossary (in English) II



**Git** is a distributed Version Control Systems (VCS) which allows multiple users to work on the same code while preserving the history of the code changes<sup>62,70</sup>. Learn more at <https://git-scm.com>.

**GitHub** is a website where software projects can be hosted and managed via the Git VCS<sup>49,70</sup>. Learn more at <https://github.com>.

**HTML** The Hyper Text Markup Language (HTML) is the text format used by the WWW<sup>3,27,69</sup>.

**IDE** An *Integrated Developer Environment* is a program that allows the user do multiple different activities required for software development in one single system. It often offers functionality such as editing source code, debugging, testing, or interaction with a distributed version control system. For Python, we recommend using PyCharm. On Apple systems, Xcode is often used.

**iOS** is the operating system that powers Apple iPhones<sup>11,63</sup>. Learn more at <https://www.apple.com/ios>.

**JavaScript** JavaScript is the predominant programming language used in websites to develop interactive contents for display in browsers<sup>21</sup>.

**JSON** *JavaScript Object Notation* is a data interchange format<sup>9,65</sup> based on JavaScript<sup>21</sup> syntax.

**Linux** is the leading open source operating system, i.e., a free alternative to Microsoft Windows<sup>2,25,61,68,76</sup>. We recommend using it for this course, for software development, and for research. Learn more at <https://www.linux.org>. Its variant Ubuntu is particularly easy to use and install.

**macOS** or Mac OS is the operating system that powers Apple Mac(intosh) computers<sup>55,63</sup>. Learn more at <https://www.apple.com/macos>.

**Matplotlib** is a Python package for plotting diagrams and charts<sup>29,30,33,48</sup>. Learn more at <https://matplotlib.org><sup>30</sup>.

**Microsoft Windows** is a commercial proprietary operating system<sup>7</sup>. It is widely spread, but we recommend using a Linux variant such as Ubuntu for software development and for our course. Learn more at <https://www.microsoft.com/windows>.

# Glossary (in English) III



- NF** The *normal forms* define guidelines for the design of relational DBs with the goal to avoid redundancy and to prevent inconsistencies and anomalies<sup>19,22,37,59,60</sup>. There are several normal forms, 1NF, second normal form (2NF), third normal form (3NF), and so on, each more restrictive than the other.
- pip** is the standard tool to install Python software packages from the PyPI repository<sup>32,50</sup>. To install a package `thepackage` hosted on PyPI, type `pip install thepackage` into the terminal. Learn more at <https://packaging.python.org/installing>.
- PostgreSQL** An open source object-relational DBMS<sup>23,45,51,64</sup>. See <https://postgresql.org> for more information.
- psql** is the client program used to access the PostgreSQL DBMS server.
- PyCharm** is the convenient Python IDE that we recommend for this course<sup>75,81,82</sup>. It comes in a free edition, so it can be downloaded and used at no cost. Learn more at <https://www.jetbrains.com/pycharm>.
- PyPI** The Python Package Index (PyPI) is an online repository that provides the software packages that you can install with `pip`<sup>5,66,74</sup>. Learn more at <https://pypi.org>.
- Python** The Python programming language<sup>28,40,42,80</sup>, i.e., what you will learn about in our book<sup>80</sup>. Learn more at <https://python.org>.
- server** In a client-server architecture, the server is a process that fulfills the requests of the clients. It usually waits for incoming communication carrying the requests from the clients. For each request, it takes the necessary actions, performs the required computations, and then sends a response with the result of the request. Typical examples for servers are web servers<sup>10</sup> in the internet as well as DBMSes. It is also common to refer to the computer running the server processes as server as well, i.e., to call it the „server computer“<sup>38</sup>.
- stderr** The *standard error stream* is one of the three pre-defined streams of a console process (together with the `stdin` and the `stdout`)<sup>36</sup>. It is the text stream to which the process writes information about errors and exceptions. If an uncaught `Exception` is raised in Python and the program terminates, then this information is written to standard error stream (`stderr`). If you run a program in a terminal, then the text that a process writes to its `stderr` appears in the console.

# Glossary (in English) IV



- stdin** The *standard input stream* is one of the three pre-defined streams of a console process (together with the `stdout` and the `stderr`)<sup>36</sup>. It is the text stream from which the process reads its input text, if any. The Python instruction `input` reads from this stream. If you run a program in a terminal, then the text that you type into the terminal while the process is running appears in this stream.
- stdout** The *standard output stream* is one of the three pre-defined streams of a console process (together with the `stdin` and the `stderr`)<sup>36</sup>. It is the text stream to which the process writes its normal output. The `print` instruction of Python writes text to this stream. If you run a program in a terminal, then the text that a process writes to its `stdout` appears in the console.
- terminal** A terminal is a text-based window where you can enter commands and execute them<sup>2,13</sup>. Knowing what a terminal is and how to use it is very essential in any programming- or system administration-related task. If you want to open a terminal under Microsoft Windows, you can `Druck auf [Win]+[R]`, dann `Schreiben von cmd`, dann `Druck auf [↵]`. Under Ubuntu Linux, `[Ctrl]+[Alt]+[T]` opens a terminal, which then runs a Bash shell inside.
- Ubuntu** is a variant of the open source operating system Linux<sup>13,26</sup>. We recommend that you use this operating system to follow this class, for software development, and for research. Learn more at <https://ubuntu.com>. If you are in China, you can download it from <https://mirrors.ustc.edu.cn/ubuntu-releases>.
- UCS** Universal Coded Character Set, see Unicode
- UML** The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems<sup>6,46,71,72</sup>
- Unicode** A standard for assigning characters to numbers<sup>31,67,73</sup>. The Unicode standard supports basically all characters from all languages that are currently in use, as well as many special symbols. It is the predominantly used way to represent characters in computers and is regularly updated and improved.

# Glossary (in English) V



**VCS** A *Version Control System* is a software which allows you to manage and preserve the historical development of your program code<sup>70</sup>. A distributed VCS allows multiple users to work on the same code and upload their changes to the server, which then preserves the change history. The most popular distributed VCS is Git.

**venv** is a Python module and tool for creating virtual environments<sup>77</sup>. Learn more at <https://docs.python.org/3/library/venv.html><sup>77</sup>.

**virtual environment** A virtual environment is a directory that contains a local Python installation<sup>43,78</sup>. It comes with its own package installation directory. Multiple different virtual environments can be installed on a system. This allows different applications to use different versions of the same packages without conflict, because we can simply install these applications into different virtual environments.

**WWW** World Wide Web<sup>3,20</sup>

**Xcode** is offers the tools for developing, testing, and distributing applications as well as an IDE for Apple platforms such as macOS and iOS<sup>57</sup>.

$\mathcal{NP}$  is the class of computational problems that can be solved in polynomial time by a non-deterministic machine and can be verified in polynomial time by a deterministic machine (such as a normal computer)<sup>24</sup>.

$\mathcal{NP}$ -hard Algorithms that guarantee to find the correct solutions of  $\mathcal{NP}$ -hard problems<sup>12,18,39</sup> have a worst-case runtime that grows exponentially with the problem scale. A problem is  $\mathcal{NP}$ -hard if all problems in  $\mathcal{NP}$  are reducible to it in polynomial time<sup>24</sup>.

$\mathbb{R}$  the set of the real numbers.